

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-87399

(43) 公開日 平成8年(1996)4月2日

(51) Int.Cl.<sup>6</sup>

識別記号

庁内整理番号

F I

技術表示箇所

G 0 6 F 7/50  
5/01  
7/00

L

8323-5E

G 0 6 F 7/ 00  
15/ 31

1 0 1 N  
S

審査請求 未請求 請求項の数9 O L (全 20 頁) 最終頁に続く

(21) 出願番号

特願平6-222950

(22) 出願日

平成6年(1994)9月19日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 山田 弘道

茨城県日立市大みか町七丁目1番1号 株

式会社日立製作所日立研究所内

(72) 発明者 村林 文夫

茨城県日立市大みか町七丁目1番1号 株

式会社日立製作所日立研究所内

(72) 発明者 山内 辰美

茨城県日立市大みか町七丁目1番1号 株

式会社日立製作所日立研究所内

(74) 代理人 弁理士 小川 勝男

最終頁に続く

(54) 【発明の名称】 冗長なシフト数予測とシフト誤り補正を用いた正規化装置

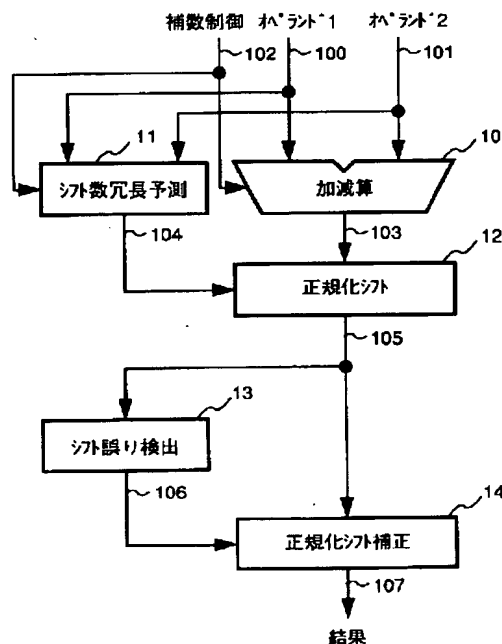
(57) 【要約】

【目的】 浮動小数点演算の主要な処理の一つである正規化処理に要する時間を短縮し、高速な演算の実行を可能とすること。

【構成】 加減算の入力データを用いて、加減算結果を正規化するために必要なシフト数をnビットの誤差範囲内で高速に予測する手段11と、この予測したシフト数と正しいシフト数の差を検出する手段13と、予測したシフト数で正規化を行った結果を0からnビットの任意にシフトする手段を設ける。これによって、シフト数の予測が誤った場合でも正しい正規化結果を得ることができる。

【効果】 加減算回路10とシフト数冗長予測回路11が同時に動作可能となるため、浮動小数点演算全体の処理時間が短縮され、高速な演算が可能になる。

図 1



## 1

## 【特許請求の範囲】

【請求項1】少なくとも1つはその最上位桁を1とする正規化表現された2つの正の2進数オペランドを入力し、加算もしくは減算オペレーションの指定によって、前記2つのオペランドを加算、または減算する加減算手段と、

前記加減算の結果最も上位に来る1の桁が前記正規化表現の最上位桁に来るように加減算結果全体をシフトする正規化シフト手段と、

前記加減算の結果最も上位に来る1の桁から前記正規化表現の最上位桁までの桁数を、前記2つの2進数オペランドを用いてn桁の誤差内で予測するシフト数冗長予測手段と、

前記正規化シフトの結果最も上位に来る1の桁が前記正規化表現の最上位桁とずれた場合に、この差を検出するシフト誤り検出手段と、

0からnの任意の数でシフトを行う正規化シフト補正手段とを有し、

前記加減算手段による加算結果は、前記シフト数冗長予測手段の予測結果を用いて前記正規化シフト手段によりシフトされ、このシフト結果は前記シフト誤り検出結果を用いて前記正規化シフト補正手段によりシフトされ、正規化結果として出力されることを特徴とする正規化装置。

【請求項2】請求項1の正規化装置において、前記シフト数冗長予測手段は1桁の誤差内で予測を行い、シフト誤り補正は、0または1桁のシフトを行うことを特徴とする正規化装置。

【請求項3】正規化表現された2つの浮動小数点オペランドを入力し、両者の指数部の比較と差計算を行い、指数部の小さいオペランドの仮数部を指数部の差だけ下位方向にシフトする桁合わせ手段と、

加算もしくは減算のオペレーションの指定と、前記入力オペランドの符号とによって、桁合わせ後の2つの仮数部を加算、または減算する加減算手段と、

前記加減算の結果最も上位に来る1の桁が前記正規化表現の最上位桁に来るように加減算結果全体をシフトする正規化シフト手段と、

前記正規化シフトの結果が前記正規化表現における所定の桁数を越える場合に、これを丸めモードに従って桁数を縮める丸め手段と、

前記加減算の結果最も上位に来る1の桁から前記正規化表現の最上位桁までの桁数を、前記桁合わせ後の2つの仮数部を用いてn桁の誤差内で予測するシフト数冗長予測手段と、

前記正規化シフトの結果最も上位に来る1の桁が前記正規化表現の最上位桁とずれた場合に、この差を検出するシフト誤り検出手段と、

0からnの任意の数でシフトを行う正規化シフト補正手段とを有し、

## 2

前記加減算手段による加算結果は、前記シフト数冗長予測手段の予測結果を用いて前記正規化シフト手段によりシフトされ、このシフト結果は前記シフト誤り検出結果を用いて前記正規化シフト補正手段によりシフトされ、正規化結果として出力されることを特徴とする正規化装置。

【請求項4】請求項3の浮動小数点演算装置において、前記シフト数冗長予測手段は1桁の誤差内で予測を行い、シフト誤り補正は、0または1桁のシフトを行うことを特徴とする浮動小数点演算装置。

【請求項5】正規化表現された3つの浮動小数点オペランドを入力し、このうち2つのオペランドの積を計算した後、3つ目のオペランドと指数部の比較と差計算を行い、指数部の小さい方の仮数部を指数部の差だけ下位方向にシフトする積和演算手段と、

加算もしくは減算のオペレーションの指定と、前記入力オペランドの符号とによって、前記積和演算後の2つの仮数部を加算、または減算する加減算手段と、

前記加減算の結果最も上位に来る1の桁が前記正規化表現の最上位桁に来るように加減算結果全体をシフトする正規化シフト手段と、

前記正規化シフトの結果が前記正規化表現における所定の桁数を越える場合に、これを丸めモードに従って桁数を縮める丸め手段と、

前記加減算の結果最も上位に来る1の桁から前記正規化表現の最上位桁までの桁数を、前記積和演算後の2つの仮数部を用いてn桁の誤差内で予測するシフト数冗長予測手段と、

前記正規化シフトの結果最も上位に来る1の桁が前記正規化表現の最上位桁とずれた場合に、この差を検出するシフト誤り検出手段と、

0からnの任意の数でシフトを行う正規化シフト補正手段とを有し、

前記加減算手段による加算結果は、前記シフト数冗長予測手段の予測結果を用いて前記正規化シフト手段によりシフトされ、このシフト結果は前記シフト誤り検出結果を用いて前記正規化シフト補正手段によりシフトされ、正規化結果として出力されることを特徴とする正規化装置。

【請求項6】請求項5の浮動小数点演算装置において、前記シフト数冗長予測手段は1桁の誤差内で予測を行い、シフト誤り補正は、0または1桁のシフトを行うことを特徴とする浮動小数点演算装置。

【請求項7】請求項1の正規化装置において、前記シフト数冗長予測手段は、加算オペレーションの場合には2つの2進数オペランドを、減算オペレーションの場合には一方のオペランドは1の補数を用いて、これらを桁ごとに排他的論理和(XOR)信号と1桁下位の論理和

(OR)信号の排他的論理和をとり、この信号を最上位桁から見て最初に1となる桁までの桁数をシフト数とす

## 3

ることにより、これが正確なシフト数に対し 1 桁の誤差内で算出され、シフト誤り補正は、0 または 1 桁のシフトを行うことを特徴とする正規化装置。

【請求項 8】請求項 3 の浮動小数点演算装置において、前記シフト数冗長予測手段は、加算オペレーションの場合には桁合わせ後の 2 つの仮数部を、減算オペレーションの場合には桁合わせ後の仮数部の一方は 1 の補数を用いて、これらを桁ごとに排他的論理和 (XOR) 信号と 1 桁下位の論理和 (OR) 信号の排他的論理和をとり、この信号を最上位桁から見て最初に 1 となる桁までの桁数をシフト数とすることにより、これが正確なシフト数に対し 1 桁の誤差内で算出され、シフト誤り補正は、0 または 1 桁のシフトを行うことを特徴とする浮動小数点演算装置。

【請求項 9】請求項 5 の浮動小数点演算装置において、前記シフト数冗長予測手段は、加算オペレーションの場合には積和演算後の 2 つの仮数部を、減算オペレーションの場合には積和演算後の仮数部の一方は 1 の補数を用いて、これらを桁ごとに排他的論理和 (XOR) 信号と

$$(-1)^s \times 2^{(e-b)} \times (1.f) \quad (1)$$

ここで、b は指数バイアスと呼ばれ、指数部 e の最大値の約半分の値である。また、小数部 f には整数の 1 が付加され、整数部と小数部を合わせて仮数部と呼ぶ。このようにして表された数値を正規化数と呼んでいる。

【0004】次に、浮動小数点演算処理について説明す

$$(-1)^0 \times 2^{(4-b)} \times (1.0011) \quad (2)$$

$$(-1)^0 \times 2^{(2-b)} \times (1.0001) \quad (3)$$

まず、減算を行う前に 2 つの数値の指数部を大きい方に一致させる。このとき指数部の小さい方の仮数部を指数部の差分だけ下位方向にシフトする。この処理を桁合わせ

$$(-1)^0 \times 2^{(4-b)} \times (0.010001) \quad (4)$$

次に、式 (2) の仮数部から式 (4) の仮数部を減算する。その結果を式 (5) に示す。

$$(-1)^0 \times 2^{(4-b)} \times (0.111011) \quad (5)$$

式 (5) の仮数部は、整数部が 0 であり正規化数とはならない。これを正規化数にするために、まず仮数部の最上位 1 (最上位桁から見て最初に 1 となる桁) を見つけ、この桁が整数部にくるように仮数部全体をシフトす

$$(-1)^0 \times 2^{(3-b)} \times (1.11011) \quad (6)$$

式 (6) の小数部は 5 桁であるため、これをフォーマットの桁数 (4 桁) に縮めるための演算を行う。これを丸

$$(-1)^0 \times 2^{(3-b)} \times (1.1110) \quad (7)$$

以上が浮動小数点演算処理の一例である。この様な浮動小数点数の演算を行う方法の従来例としては、特開平 2-232723 号に述べられている方法がある。この方法を用いた演算装置を図 16 を用いて説明する。

【0010】桁合わせ回路 15 は、加減算を施すべき 2 つの浮動小数点数オペランド 110、111 を入力し、両方の指数部を大きい方の値に一致させ、指数部の小さい方の仮数部を指数部の差だけ下位方向にシフトする。

## 4

1 桁下位の論理和 (OR) 信号の排他的論理和をとり、この信号を最上位桁から見て最初に 1 となる桁までの桁数をシフト数とすることにより、これが正確なシフト数に対し 1 桁の誤差内で算出され、シフト誤り補正は、0 または 1 桁のシフトを行うことを特徴とする浮動小数点演算装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は正規化処理を必要とする浮動小数点数の加算、積和演算を行う浮動小数点演算装置に関する。

【0002】

【従来の技術】浮動小数点数を計算機上で表現する方法としては、IEEE-754 で定められた標準形式が一般に用いられる。符号 s、指数部 e、小数部 f からなるデータで表現された浮動小数点数は、以下の式で表される数値をとる。

【0003】

$$(-1)^s \times 2^{(e-b)} \times (1.f) \quad (1)$$

る。簡単のため符号 1 ビット、指数部 3 ビット、小数部 4 ビットとして、式 (2)、(3) で表される 2 つの浮動小数点数の減算を考える。

【0005】

$$(-1)^0 \times 2^{(4-b)} \times (1.0011) \quad (2)$$

$$(-1)^0 \times 2^{(2-b)} \times (1.0001) \quad (3)$$

せと呼び、式 (3) を式 (4) へ変換する。

【0006】

$$(-1)^0 \times 2^{(4-b)} \times (0.010001) \quad (4)$$

【0007】

$$(-1)^0 \times 2^{(4-b)} \times (0.111011) \quad (5)$$

る。次に、指数部から仮数部をシフトした数を減ずる。この処理を正規化と呼び、式 (5) を式 (6) に変換する。

【0008】

$$(-1)^0 \times 2^{(3-b)} \times (1.11011) \quad (6)$$

め処理と呼ぶ。式 (6) を丸めた例を式 (7) に示す。

【0009】

$$(-1)^0 \times 2^{(3-b)} \times (1.1110) \quad (7)$$

加減算回路 10 は、オペレーションの指示 112 と入力オペランド 110、111 の符号の組み合わせ、指数部、仮数部の大小関係等に対応して桁合わせ後の仮数部 113、114 の加減算を行う。シフト数計算回路 90 は、加減算結果 103 の最上位 1 の桁から整数部までの桁数をシフト数 180 として計算する。正規化シフト回路 12 は、シフト数 180 を用いて、加減算結果 103 をシフトする。丸め回路 16 は、正規化シフト結果 105 を

## 5

フォーマットの桁数に縮めるための演算を行う。指数部補正回路17は、桁合わせ回路15が出力する指数部115からシフト数180を減じた値116とこれに1を加えた値117を出力する。指数部選択回路18は、丸め結果115に応じて、補正後の指数部116, 117のいずれかを選択する。以上の処理によって浮動小数点数の演算結果119が生成される。

【0011】

【発明が解決しようとする課題】しかしながら、従来の演算装置では加減算結果を用いて正規化シフト数を計算していたため、正規化処理に時間がかかり高速化の妨げとなっていた。

【0012】本発明の目的は、正規化処理をより高速に行うことのできる演算方式を提供することにある。また、本発明の他の目的は、正規化処理を高速化することにより、浮動小数点数の加減算、積和演算をより高速に行うことのできる浮動小数点演算装置を提供することに

$$A = A[0] A[1] \cdots A[n]$$

$$B = B[0] B[1] \cdots B[n]$$

次に、A, Bに式(10), (11)の2つの演算を施す。

$$X = A \wedge B$$

$$P = A \mid \mid B$$

ここで、“ $\wedge$ ”は、排他的論理和の演算子、“ $\mid \mid$ ”は、論理和の演算子である。次に、X, Pに式(12)

$$Z = X \ll (P < < 1)$$

ここで、“ $\ll$ ”は、シフト演算子で、“ $P < < 1$ ”は、Pを左に1ビットシフトするという意味である。但し、シフト前の最上位桁を越える桁は無視する。式(1

$$A = 01011001$$

$$B = 10111101$$

として、これらを式(10), (11)に代入すると、

$$X = 11100100$$

$$P = 11111101$$

となる。これらを式(12)に代入して、

$$Z = 00011110$$

という結果を得る。これは、AとBを加算した結果、上位3桁は“0”が連続し、4桁目で初めて“1”となる

$$S = 00010110$$

となり、式(17)と式(18)で最上位1の桁が一致

$$A = 01010001$$

$$B = 10111101$$

$$X = 11101100$$

$$P = 11111101$$

$$Z = 00010110$$

$$S = 00001110$$

となり、式(23)と式(24)で最上位1の桁が一致しない。このように、式(12)の演算結果の最上位1の桁は、加算結果のそれと等しいか、1桁上位のいずれかとなる。これは、式(12)が、加算結果の最上位桁か

## 6

ある。

【0013】

【課題を解決するための手段】上記目的を達成するために本発明の浮動小数点演算装置では、正規化シフト数を加減算の入力データを用いてnビットの誤差内で予測するシフト数冗長予測手段と、予測したシフト数と正確なシフト数との差分を検出するシフト誤り検出手段と、予測したシフト数を用いて正規化シフトした結果をシフト誤りの数だけシフトするnビットシフタを備えている。

【0014】ここで、シフト数冗長予測手段に用いる演算アルゴリズムについて説明する。本アルゴリズムは、加減算処理に必要な長いビットにわたる桁上げ伝播および論理段数を必要とせず、たかだか2ビット間の信号伝播と論理段数2段で、加減算結果の最上位桁から連続する“0”を1ビットの誤差内で求めることができる。まず、補数処理を施した後のデータを、A, Bとする。

【0015】

$$\cdots (8)$$

$$\cdots (9)$$

【0016】

$$\cdots (10)$$

$$\cdots (11)$$

の演算を施す。

【0017】

$$\cdots (12)$$

2)のZが、加算結果の最上位桁から連続する“0”を表す。実際に、具体的な数値を用いて説明する。

【0018】

$$\cdots (13)$$

$$\cdots (14)$$

$$\cdots (15)$$

$$\cdots (16)$$

$$\cdots (17)$$

ことを示す。実際に、AとBを加算した結果は、

$$\cdots (18)$$

40 している。もう1つの例を示すと、

$$\cdots (19)$$

$$\cdots (20)$$

$$\cdots (21)$$

$$\cdots (22)$$

$$\cdots (23)$$

$$\cdots (24)$$

ら連続する“0”のうち最後の“0”以外を正確に計算することができるからである。これについて、証明を行う。

【0019】式(8), (9)を加算した結果のi桁目をS

7

[i]、i+1桁目をS[i+1]とすると、

$$S[i] = X[i] \wedge C[i+1] \quad \dots (25)$$

$$= X[i] \wedge (G[i+1] \mid (P[i+1] \& C[i+2])) \quad \dots (26)$$

$$S[i+1] = X[i+1] \wedge C[i+2] \quad \dots (27)$$

ここで、X[i+1]、G[i+1]、P[i+1]は、それぞれi+1桁目の2入力の排他的論理和、論理積、論理和であり、C[i+1]は、i+1桁目からi桁目への桁上がりである。また、記号“^”、“&”、

$$X[i+1] \wedge C[i+2] = 0 \quad \dots (28)$$

$$X[i+1] = C[i+2] \quad \dots (29)$$

が導かれる。次に、式(29)を式(26)に代入し、

$$X[i] \wedge (G[i+1] \mid (P[i+1] \& X[i+1])) = 0 \quad \dots (30)$$

が導かれ、ここで、

$$G[i+1] \mid (P[i+1] \& X[i+1]) = P[i+1] \quad \dots (31)$$

であることから、式(30)はさらに式(32)のように変形される。

$$X[i] \wedge P[i+1] = 0$$

すなわち、S[i]とS[i+1]がいずれも“0”の場合には、式(32)が成立する。これを拡張して考えると、加算結果の最上位桁からi+1桁目までがすべて“0”の場合には、最上位桁からi桁目について、式(32)が成立することになる。これは、式(12)の最上位桁からi桁目がすべて“0”となるということに等しい。以上により、式(12)の結果を用いれば、1桁の誤差内で加算結果の最上位1が予測可能なことが証明された。

#### 【0021】

【作用】上記のように、正規化シフト数を加減算の入力データを用いてnビットの誤差内で予測することにより、正規化シフト数計算と加減算とが並列に実行可能になる。また、シフト誤り検出と正規化シフト補正により、正しい正規化結果を生成可能になる。

#### 【0022】

【実施例】図2は、本発明を用いた計算機システムの例である。プロセッサ21、メモリ22、ハードディスク23からなる計算機が複数台、総合ネットワーク20を介して接続される。係る計算機システムにおいて、本発明は数値計算処理を司るプロセッサ21に適用される。

【0023】図3は、プロセッサ21の例である。データ及び命令を高速にアクセスできるように、データキャッシュ30、命令キャッシュ33を内蔵する。仮想アドレスから物理アドレスへのメモリアドレス変換は、データTLB31と命令TLB32を用いて行う。この制御はメモリ制御ユニット34が行う。整数演算は、汎用レジスタ35、ALU36、ALU37で行う。命令アドレスの計算は、アドレス加算器38を用いて行う。浮動小数点演算は、浮動小数点レジスタ40、浮動小数点加算器41、浮動小数点乗算器42、浮動小数点除算器43で行う。これらの制御は命令制御ユニット39が行う。

8

“|”は、それぞれ排他的論理和、論理積、論理和のビット演算子を意味する。ここで、S[i]とS[i+1]がいずれも“0”であるとしたときの式(26)の変形を行う。まず、式(27)より、

$$\dots (28)$$

$$\dots (29)$$

【0020】

$$\dots (32)$$

科学技術計算やコンピュータ・グラフィックス等の計算機利用分野では、非常に高い浮動小数点演算性能が必要である。特に、浮動小数点加算は浮動小数点演算命令の中でも使用頻度が高く、高速演算が可能な浮動小数点加算器は重要である。本発明は、係る浮動小数点加算器41に適用される。

【0024】図4は、浮動小数点加算器41の実施例である。図4において、10は加減算回路、11はシフト数冗長予測回路、12は正規化シフト回路、13はシフト誤り検出回路、14は正規化シフト補正回路、15は桁合わせ回路、16は丸め回路、17は指数部補正回路、18は指数部選択回路である。本実施例の演算装置は、2つの入力オペランド(110、111)をオペレーションの指示(112)に従い加算または減算し、正規化および丸めの処理を行った結果(119)を出力する。

【0025】これより以降、IEEE-754倍精度フォーマットを例に説明を行うが、本発明の効果はこれに限られるわけではなく、単精度フォーマットや4倍精度フォーマットにも、またIEEEフォーマット以外にも広く利用可能である。

【0026】以下、本実施例について図表を用いて詳細に説明する。

【0027】まず、桁合わせ回路15について説明する。図5は、桁合わせ回路15の入力オペランド(110、111)のフォーマットと、桁合わせ結果の仮数部fa(113)、fb(114)を示す。表1は、桁合わせ回路15の機能を示す。桁合わせ後の指数部ec(115)は、2つの入力オペランド(110、111)の指数部e1、e2のうち大きい方の値をとる。桁合わせ後の仮数部fa(113)は、指数部の大きい方の小数部に整数値1を付加した値である。桁合わせ後のもう一方

の仮数部  $f_b$  (114) は、指数部の小さい方の小数部に整数値 1 を付加した後、これを指数部の差だけ下位方向にシフトした値である。ここで、 $f_b$  (114) の最下位桁の信号  $b_{55}$  は、シフトによって 55 桁目から桁落ちした全信号の OR である。本実施例が対象とする加

表 1

桁合わせ出力	$e1 \geq e2$ の場合	$e1 < e2$ の場合
指数部 $ec$ (115)	$e1$	$e2$
仮数部 $fa$ (113)	1. $f1$	1. $f2$
仮数部 $fb$ (114)	1. $f2$ を指数部の差 ( $e1 - e2$ ) だけ、下位方向にシフト。但し、小数部 55 ビット目 ( $b_{55}$ ) は、55 ビット目以下の信号全ての OR。	1. $f1$ を指数部の差 ( $e2 - e1$ ) だけ、下位方向にシフト。但し、小数部 55 ビット目 ( $b_{55}$ ) は、55 ビット目以下の信号全ての OR。

【0029】オペレーションの指示 (112) と入力オペランド (110, 111) の符号  $s1$ ,  $s2$  の値によって、桁合わせ後の仮数部の処理が実質加算 ( $fa + fb$ )、実質減算 A ( $fa - fb$ )、実質減算 B ( $fb - fa$ ) のいずれかに決まる。この処理を加減算回路 10 で行わせるために、桁合わせ回路で補数制御信号 116 を生成する。表 2 に示すのは、補数制御信号 116 の一例で、117, 118, 119 の 3 信号より成る。117

表 2

演算の区分	補数制御 (116)			条件
	117	118	119	
実質加算 ( $fa + fb$ )	0	0	0	オペレーション=加算かつ $s1=s2$ または、オペレーション=減算かつ $s1 \neq s2$
実質減算 A ( $fa - fb$ )	0	1	1	オペレーション=加算かつ $s1 \neq s2$ または、オペレーション=減算かつ $s1=s2$ で、 ( $e1=e2$ ) かつ ( $f1 < f2$ ) 以外
実質減算 B ( $fb - fa$ )	1	0	1	オペレーション=加算かつ $s1 \neq s2$ または、オペレーション=減算かつ $s1=s2$ で、 ( $e1=e2$ ) かつ ( $f1 < f2$ )

【0031】次に、加減算回路 10 について説明する。図 6 は、加減算回路 10 の一例で、LSI によく用いられる桁上げ先見加算器のブロック図である。補数回路 50, 51 は、実質減算の場合に仮数部  $fa$  (113) または仮数部  $fb$  (114) の補数をとる回路である。補

算または減算に限って言えば、このように桁合わせの桁数を小数部 55 桁までに制限しても演算精度上何ら問題ない。

【0028】

【表 1】

は、実質減算 B の場合に 1 となる信号で、桁合わせ後の仮数部  $fa$  の補数をとるために用いる。118 は、実質減算 A の場合に 1 となる信号で、桁合わせ後の仮数部  $fb$  の補数をとるために用いる。119 は、実質減算 A または実質減算 B の場合に 1 となる信号で、加減算回路 10 で減算を行うときに補数の 1 として用いる。

【0030】

【表 2】

数制御は、それぞれ信号 117, 118 を用いて行う。補数回路 50, 51 の出力 120, 121 は、まず数ビットを単位とするいくつかのフィールドに分けて、これらフィールド毎に加算を行う。52 は、下位からの桁上がりがないと仮定した加算回路、53 は、下位からの桁

上がりがあると仮定した加算回路、54は、補数制御信号119を最下位桁への桁上がりとして用いる加算回路である。ここで、図中左が上位桁側であるとする。加算回路52、53、54は、各フィールドの和122と桁上がり123を計算する。このうち桁上がり123を桁上げ伝搬回路55に入力し、各フィールドへの正しい桁上がり124が計算される。56は、加算結果選択回路で、桁上がり124を用いて、フィールド毎に加算回路52または加算回路53の和122のうち正しい方を選択する。以上の処理によって、加減算結果S1 S0, s1...s55が求まる。

【0032】次に、シフト数冗長予測回路11について説明する。図7は、シフト数冗長予測回路11の一例である。桁合わせ回路15の出力する仮数部fa(113), fb(114)および補数制御信号117, 118を用いて、加減算回路10の出力103の最上位桁から見て最初に1となる桁から整数部までの桁数を1ビットの誤差内で予測する。

【0033】図7において、補数回路60, 61は、実質減算の場合に仮数部fa(113)または仮数部fb(114)の補数をとる回路である。補数制御は、それぞれ信号117, 118を用いて行う。連続0の検出回路62は、補数回路60, 61の出力信号130, 131を加算した場合に、その結果の最上位桁から下位方向に連続する0を求める回路である。図8に、連続0の検出回路62の詳細回路図を示す。この回路は、補数回路60, 61の出力信号130, 131を各ビット毎に、OR(論理和)およびXOR(排他的論理和)をとった後、XOR信号と1ビット下のOR信号のXORをとる。この結果生成される55ビットの信号z0 z1 z2...z54が連続0検出結果132となる。この信号は、z0が小数点のすぐ左の整数部に対応し、z1...z54が小数部1桁目から54桁目に対応する。例えば、z0...z5がすべて0で、z6が1である場合には、加算結果の最上位1は小数部第6桁目だと考え、シフト数を6とする。但し、この値は1ビットの誤差を含んでおり、この例の場合には加算結果の最上位1は小数部第6桁目ではなく第5桁目であり、シフト数は5とするのが正しいというように、シフト数を1だけ多く予測してしまう場合がある。これは、正確な連続0検出のためには連続する0の最後の桁に正しい桁上がりを入力する必要があるのに、図8の回路では正しい桁上りの代わりにすぐ下の桁の信号のORを用いるためである。このように1ビットの誤差があるものの、図8に示す回路はわずかXOR回路2段で、加算結果の連続0検出が可能である。

【0034】再び図7に戻って、シフト数冗長予測回路11の説明を続ける。図7において、63は優先順位判定回路、64はバイトシフト数計算回路、65はビットシフト数計算回路である。本実施例における正規化シフ

ト回路12として、後述するように加減算結果103をまずバイト単位にシフトし、次にビット単位のシフトを行うといった2段シフトの例を挙げる。このようなシフトでは、シフト数をバイト数とビット数に分けて表し、バイト単位のシフトとビット単位のシフトにそれぞれ入力する必要がある。バイトシフト数計算回路64は、バイト単位のシフト数を意味する信号147を生成する。ビットシフト数計算回路65は、ビット単位のシフト数を意味する信号148を生成する。連続0検出結果132は、最上位桁から8ビットずつ7つの優先順位判定回路63に入力される。図9に、最も上位側の優先順位判定回路63の例を示す。残りの優先順位判定回路もこれと同一の構成である。連続0検出結果132の上位8ビットの信号z0...z7を入力し、ブロックOR信号133とブロック内優先順位信号139を生成する。ブロックOR信号133は、8ビットの入力信号のORで、これが1の場合には入力信号に1が存在することを意味する。ブロック内優先順位信号139は、8ビットの入力信号のうち最上位に最も近い1の桁のみを1とし、それ以外をすべて0とした信号である。但し、入力信号の上位7桁が全て0の場合は、図9に示すように入力信号の8桁目の値に係わらず、この桁を1とみなしても良い。これは、入力8ビットが全て0の場合には、このブロック内優先順位信号139は後の計算に用いられないためである。

【0035】図10に、バイトシフト数計算回路64の例を示す。優先順位判定回路63のうち、上位側6つが出力するブロックOR信号(133~138)がバイトシフト数計算回路64に入力される。この回路は、最上位に最も近い1の信号のみを1とし、それ以外を全て0とするように動作する。信号133~138が全て0の場合には、最も下位の優先順位判定回路64のブロックORが1であるとみなす。このようにして生成された7ビットの信号y0 y1...y6がバイトシフト数を示す信号147となる。

【0036】再び図7に戻って、ビットシフト数計算回路65の説明を行う。7つの優先順位判定回路63が出力するブロック内優先順位信号139~146を、バイトシフト数147で選択し、ビットシフト数148とする。

【0037】以上のようにして求められたバイトシフト数147およびビットシフト数148が、シフト数冗長予測結果104となる。

【0038】次に、正規化シフト回路12について説明する。図11は、正規化シフト回路12の一例で、バイト単位のシフトとビット単位のシフトの2段でシフトを行う回路のブロック図である。70は、7入力から1つを選択する回路で、これを57個使用して57ビットのバイトシフトを構成している。各選択回路70は、加減算回路10の出力信号103のうち、そのビットに当た

10

20

30

40

50

る信号を先頭に8の整数倍だけ下位にある信号を7本入力する。入力に用いる信号が無い場合は0を入力する。入力信号の選択は、シフト数冗長予測回路11の出力するバイトシフト数147を用いて行い、57ビットの信号Y1 Y0. y1 y2...y55(150)を生成する。71は、8入力から1つを選択する回路で、これを57個使用して57ビットのビットシフタを構成している。各選択回路71は、バイトシフタの出力信号150のうち、そのビットに当たる信号を先頭に連続する8本を入力する。入力に用いる信号が無い場合は0を入力する。入力信号の選択は、シフト数冗長予測回路11の出力するビットシフト数148を用いて行い、57ビットの正規化シフト結果N1 N0. n1 n2...n55(105)を生成する。

【0039】次に、図12を用いてシフト誤り検出回路13と正規化シフト補正回路14について説明する。正規化シフト回路12の出力信号105を入力し、シフト誤り検出回路13の結果信号106によって、信号105をそのままあるいは1ビット下位方向にシフトし、正規化シフト補正結果107として出力する。72は、2入力から1つを選択する回路で、これを56個使用して56ビットの1ビットシフタを構成している。各選択回路72は、正規化シフト結果信号105のうち、そのビットに当たる信号とその1ビット上位の信号の2本を入力する。入力信号の選択は、正規化シフト結果信号105の最上位桁の信号であるN1を用いて行う。すなわち、この1ビットの信号N1そのものがシフト誤り検出結果信号106に等しい。N1が0の場合には、正規化シフトが正しく行われており、正規化シフトの補正は必要ない。N1が1の場合には、正規化シフトが誤って1ビット余計に行われているため、正規化シフトの補正が必要である。このようにして、正しい正規化シフト結果U0. u1 u2...u55(107)が得られる。

【0040】次に、丸め回路16について説明する。図13は丸め回路16の一例を示すブロック図である。73は、正規化シフト補正結果107のうち上位53桁のU0. u1 u2...u52に1を加算した値151を生成するインクリメント回路である。74は、丸め判定回路で、信号107のうち下位の信号u52...55を使って、丸めによる桁上がり信号152を生成する。75は、正規化シフト補正結果107とインクリメント結果151を入力し、丸めによる桁上がり信号152がない場合には信号107を、ある場合には信号151を選択し丸め結果R1 R0. r1...r52(115)とする、丸め

選択回路である。丸め結果の整数部を2桁求めるのは、正規化シフト補正結果107が全て1の場合に、丸めによる桁上がりが発生し、最上位1の桁が整数部1桁目から2桁目にずれる場合があるためである。この場合、丸め結果115の小数部は全て0であるから、最上位1を小数点のすぐ左に戻すためのシフトは必要ない。

【0041】次に、指数部補正回路17について説明する。図14は、指数部補正回路17の一例を示すブロック図である。図14において、76は、シフト数冗長予測結果104のうちバイトシフト数を意味する信号147を入力し、これを3ビット信号160に符号化する回路である。77は、シフト数冗長予測結果104のうちビットシフト数を意味する信号148を入力し、これを3ビット信号161に符号化する回路である。信号160の下に信号161を合わせると、シフト数冗長予測信号104を2進数とした値162が生成される。78は、桁合わせ回路15の出力する桁合わせ後の指数115からシフト数冗長予測信号の2進数値162を減算した値163とこれに1を加えた値164を出力する回路である。最後に、指数部選択回路18について説明する。この回路は、シフト誤り検出信号106がある場合または、丸め結果115の信号R1が1の場合に、指数部補正回路の出力信号164を、そうでない場合に信号163を選択し結果の指数部118とする指数部選択回路である。以上説明した処理によって生成される信号118と信号115をもって、浮動小数点加算器の結果119とする。

【0042】図4の実施例における、シフト数冗長予測の具体例を表3および表4に示す。いずれの表も、桁合わせ回路15の出力する仮数部113から114を減算した場合に、正規化シフト回路12の結果105が生成されるまでの主要信号を示している。表3は、シフト数冗長予測が正しい例で、加減算回路10の結果信号103の最上位1が小数点第1桁目にあり、シフト数冗長予測信号147, 148の示す値は1(0バイト, 1ビット)となっている。従って、正規化シフト結果105は正規化数になっている。一方表4は、シフト数冗長予測が誤る例で、加減算回路10の結果信号103の最上位1が整数部第1桁目にあり、シフト数冗長予測信号147, 148の示す値は1(0バイト, 1ビット)となっている。従って、正規化シフト結果105は正規化数とはならない。

【0043】

【表3】



表 3

ブロック名	信号名	信号値
桁合わせ回路15	113	01.0011...
	114	00.0100...
	116 (117, 118, 119)	011
加減算回路10 (シフト数冗長予測回路11)	120 (130)	01.0011...
	121 (131)	11.1011...
加減算回路10	103	00.111...
シフト数冗長予測回路11	132	01110...
	147	1000000 (=0)
	148	01000000 (=1)
正規化シフト回路12	105	01.11...

【0044】

20 表 4

ブロック名	信号名	信号値
桁合わせ回路15	113	01.1011...
	114	00.0100...
	116 (117, 118, 119)	011
加減算回路10 (シフト数冗長予測回路11)	120 (130)	01.1011...
	121 (131)	11.1011...
加減算回路10	103	01.011...
シフト数冗長予測回路11	132	01011...
	147	1000000 (=0)
	148	01000000 (=1)
正規化シフト回路12	105	10.11...

【0045】図15には、本発明を用いた別の実施例である浮動小数点積和演算器のブロック図を示す。この装置は、3入力オペランド170、171、172をオペレーションの指示173に従い、オペランド170と171の積にオペランド172を加算もしくは減算し、正規化および丸めを施して結果119を生成する。図4の実施例装置と異なる点は、桁合わせ回路15が積和演算回路80に置き換えられている点である。積和演算回路80の出力する2つの仮数部174、175は、加減算回路10にて加算または減算が行われ、174冗長シフト数予測回路11にて正規化シフト数の予測が行われ

40 る。

【0046】

【発明の効果】本発明の効果を図17および図18を用いて説明する。

【0047】図17は、図4の実施例装置と図16の従来装置について、演算時間を決定する要因であるクリティカルパスの比較を行った図である。同図(a)は、従来装置のクリティカルパスを示す。図16において、桁合わせ回路15の出力信号113をパスの始点、正規化シフト回路12の出力信号105をパスの終点とした。

50 図17(b)は、図4の実施例装置のクリティカルパス

を示す。図 4 において、桁合わせ回路 15 の出力信号 113 をパスの始点、正規化シフト補正回路 14 の出力信号 107 をパスの終点とした。図 17 (a) では、加減算、シフト数計算、正規化シフトが逐次処理となっているのに対し、同図 (b) では、加減算とシフト数計算が並列処理となって論理段数が大きく減少している。正規化シフトの後に、シフト誤り検出と正規化シフト補正という新たな処理が必要になるものの、クリティカルパス全体としては高速化されている。

【0048】図 18 は、図 4 の実施例と図 16 の従来装置について、演算時間の比較を行った図である。横軸は、演算時間を表す。同図 (b) からわかるように、シフト数計算が加減算と同時に進められるようになったため、正規化に要する時間が従来装置の半以下になっている。加減算と正規化の処理で比較すると従来に比べ 30 % 以上、浮動小数点加算の処理全体で比較すると 15 % 以上の高速化の効果が得られている。

【0049】また、本発明を実施するにあたっては、従来装置に比べ、シフト数冗長予測回路中の連続 0 検出と、正規化シフト補正回路に要する論理回路が新たなハードウェアとして必要となるが、これらに必要なゲート数は、浮動小数点加算器全体の 1 ~ 2 % 程度であり、ゲート数の増加も少なく抑えることができる。

#### 【図面の簡単な説明】

【図 1】本発明の特徴を最もよく表す正規化装置のブロック図である。

【図 2】本発明を用いた計算機システムの一例である。

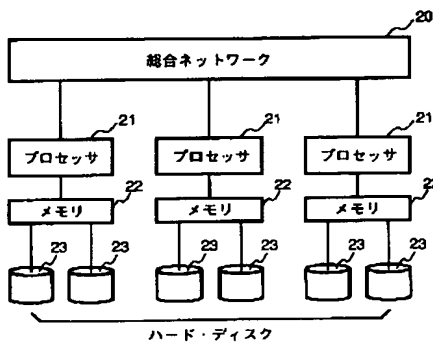
【図 3】図 2 のプロセッサ 21 のブロック図で、本発明を適用したプロセッサの一例である。

【図 4】本発明の実施例の一つである浮動小数点加算器のブロック図である。

【図 5】浮動小数点数の表現方法の一例である。

【図 2】

図 2



【図 6】図 4 の加減算回路 10 の詳細ブロック図である。

【図 7】図 4 のシフト数冗長予測回路の詳細ブロック図である。

【図 8】図 7 の連続 0 検出回路 62 の詳細論理図である。

【図 9】図 7 の優先順位判定回路 63 の詳細論理図である。

【図 10】図 7 のバイトシフト数計算回路 64 の詳細論理図である。

【図 11】図 4 の正規化シフト回路 12 のブロック図である。

【図 12】図 4 の正規化シフト補正回路 14 のブロック図である。

【図 13】図 4 の丸め回路 16 のブロック図である。

【図 14】図 4 の指数部補正回路 17 のブロック図である。

【図 15】本発明の実施例の一つである浮動小数点積和演算器のブロック図である。

【図 16】従来の浮動小数点加算器のブロック図である。

【図 17】浮動小数点加算器のクリティカルパスを従来装置と本発明の実施例装置とで比較した図である。

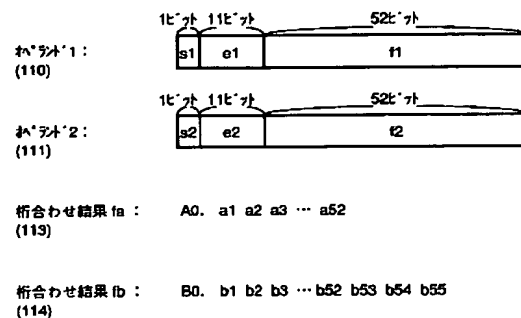
【図 18】浮動小数点加算器の演算時間を従来装置と本発明の実施例装置とで比較した図である。

#### 【符号の説明】

10…加減算回路、11…シフト数冗長予測回路、12…正規化シフト回路、13…シフト誤り検出回路、14…正規化シフト補正回路、15…桁合わせ回路、16…丸め回路、17…正規化シフトに伴う指数部補正回路、18…指数部選択回路。

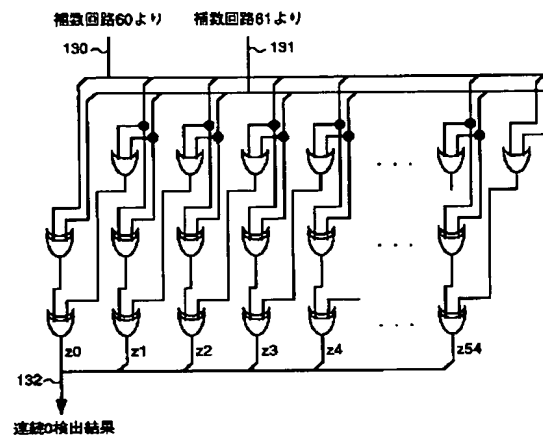
【図 5】

図 5



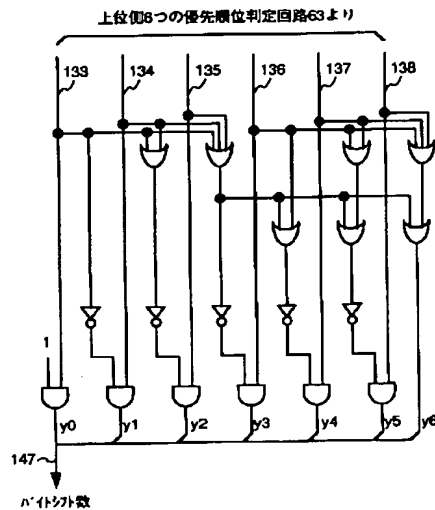
【図 8】

8



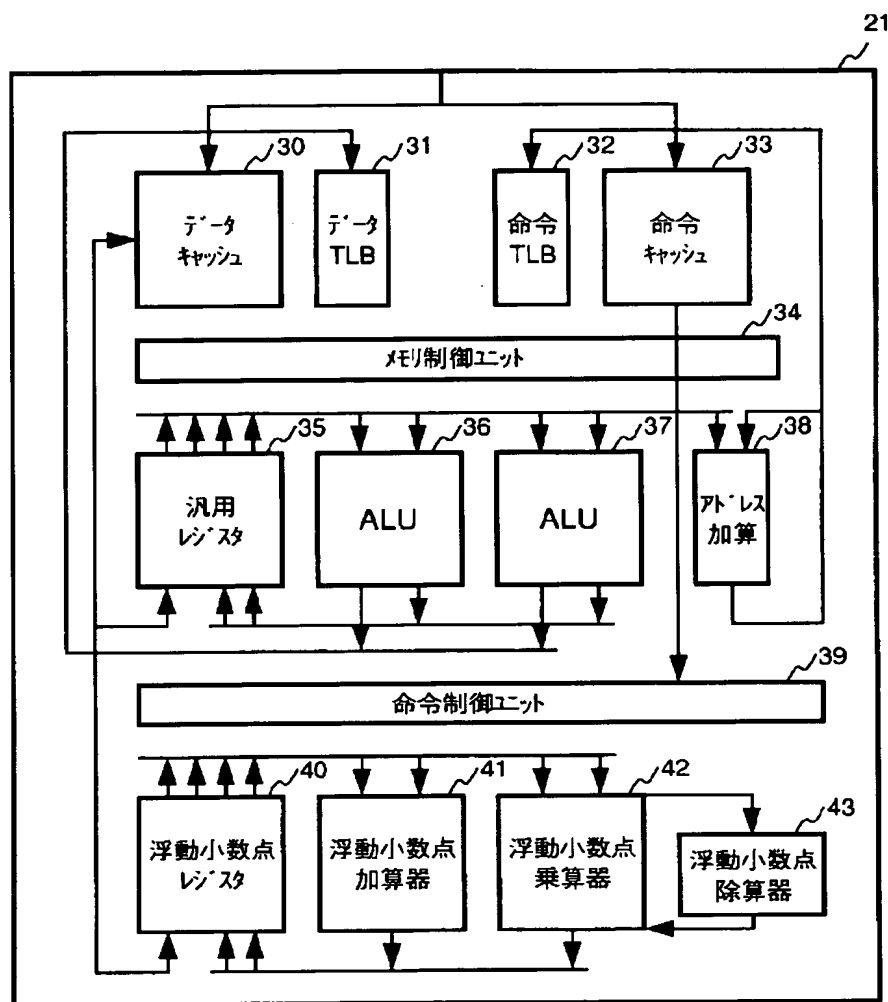
【図 10】

**圖 10**



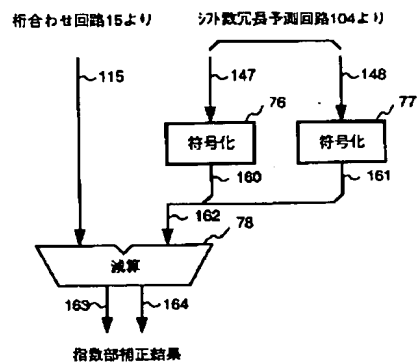
【図 3】

図 3



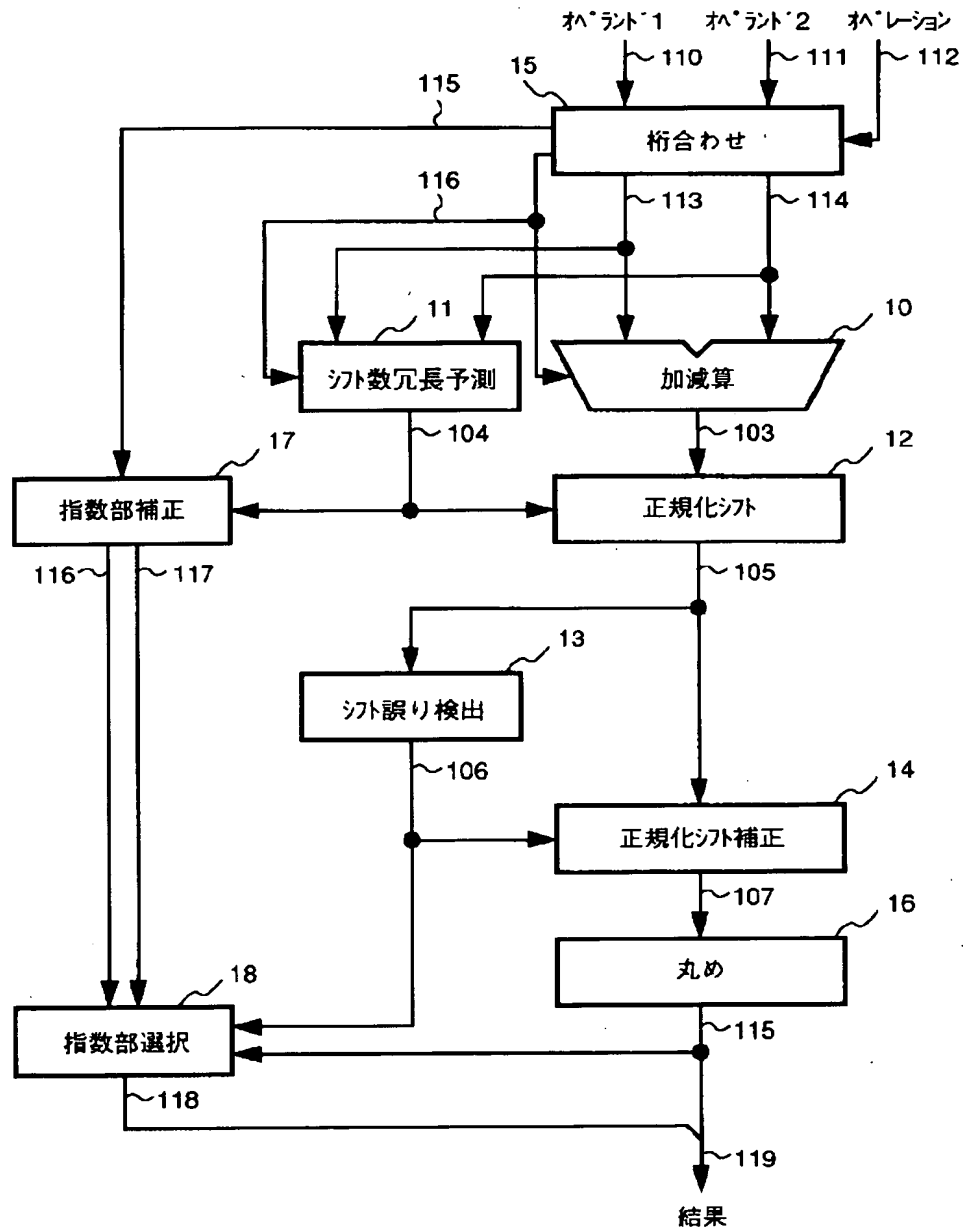
【図 14】

図 14



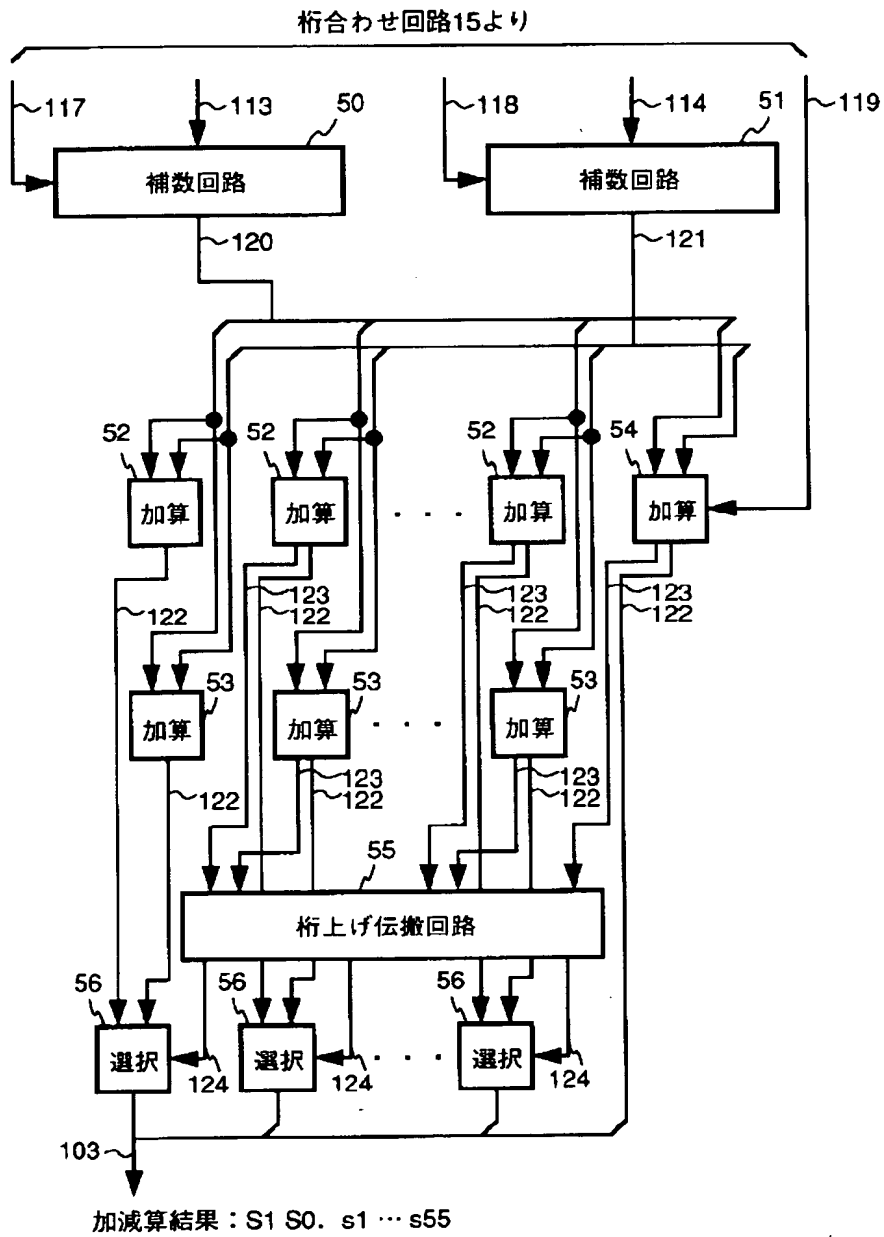
【図 4】

図 4



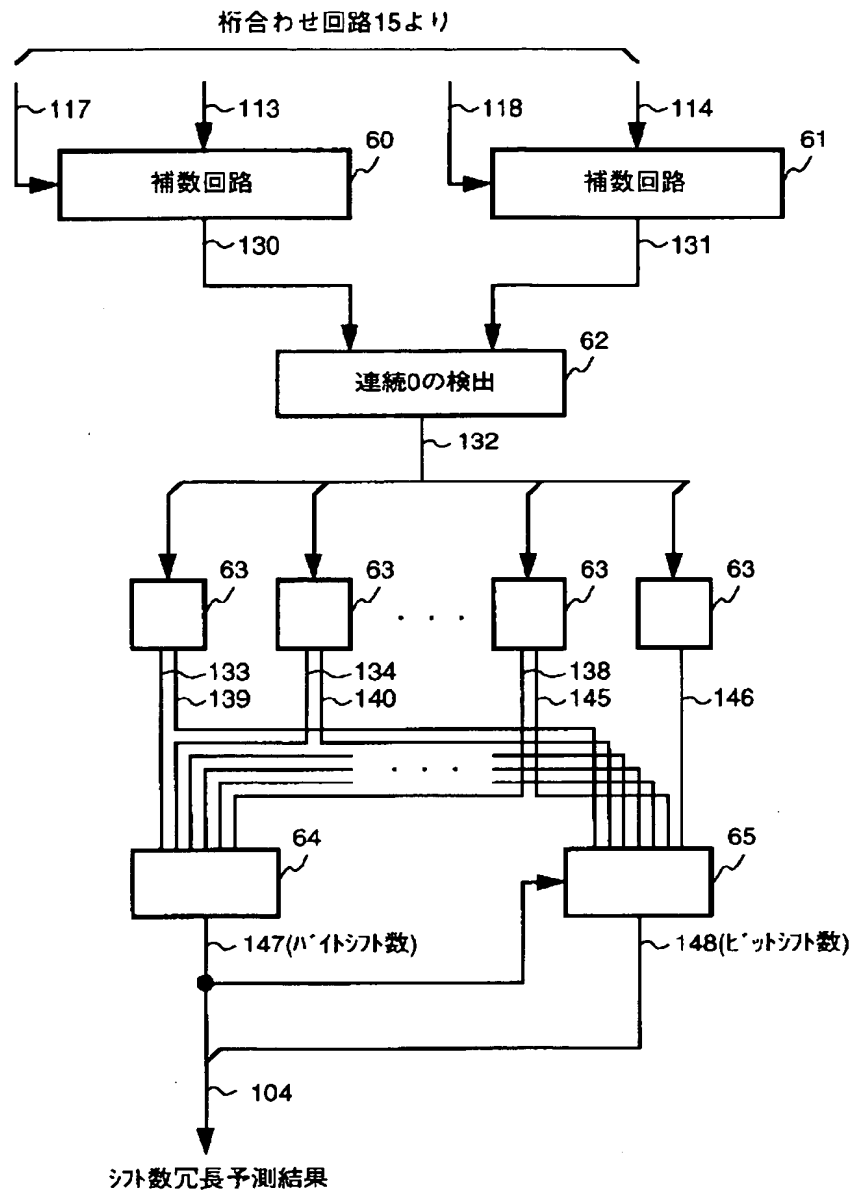
【図 6】

図 6



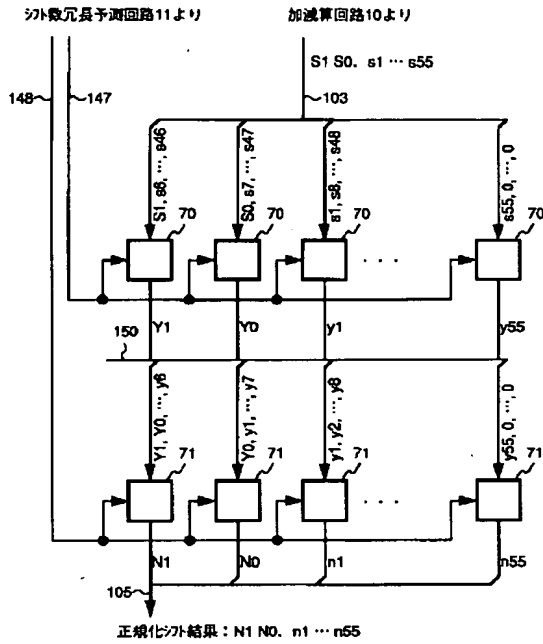
【図 7】

図 7



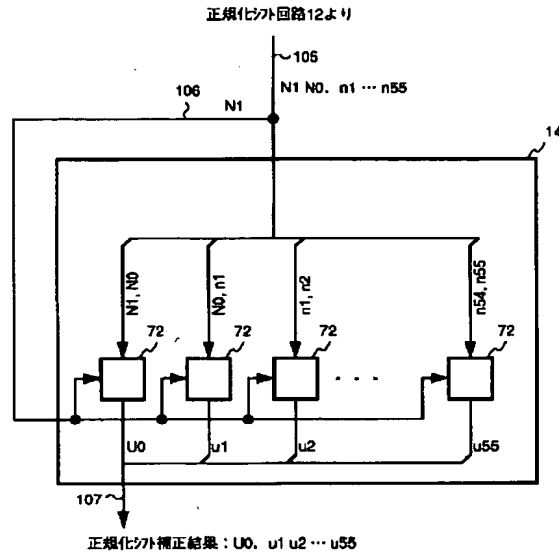
【図 11】

図 11



【図 12】

図 12



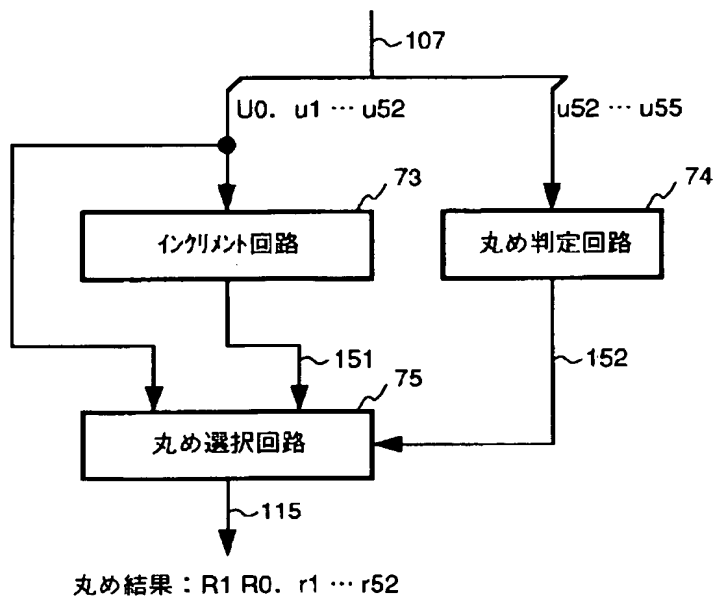
【図 18】

図 18

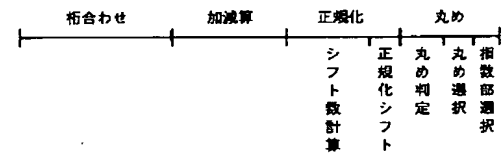
【図 13】

図 13

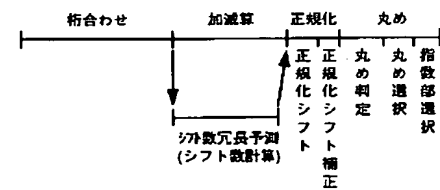
正規化シフト補正回路14より



(a) 従来の演算装置の処理時間



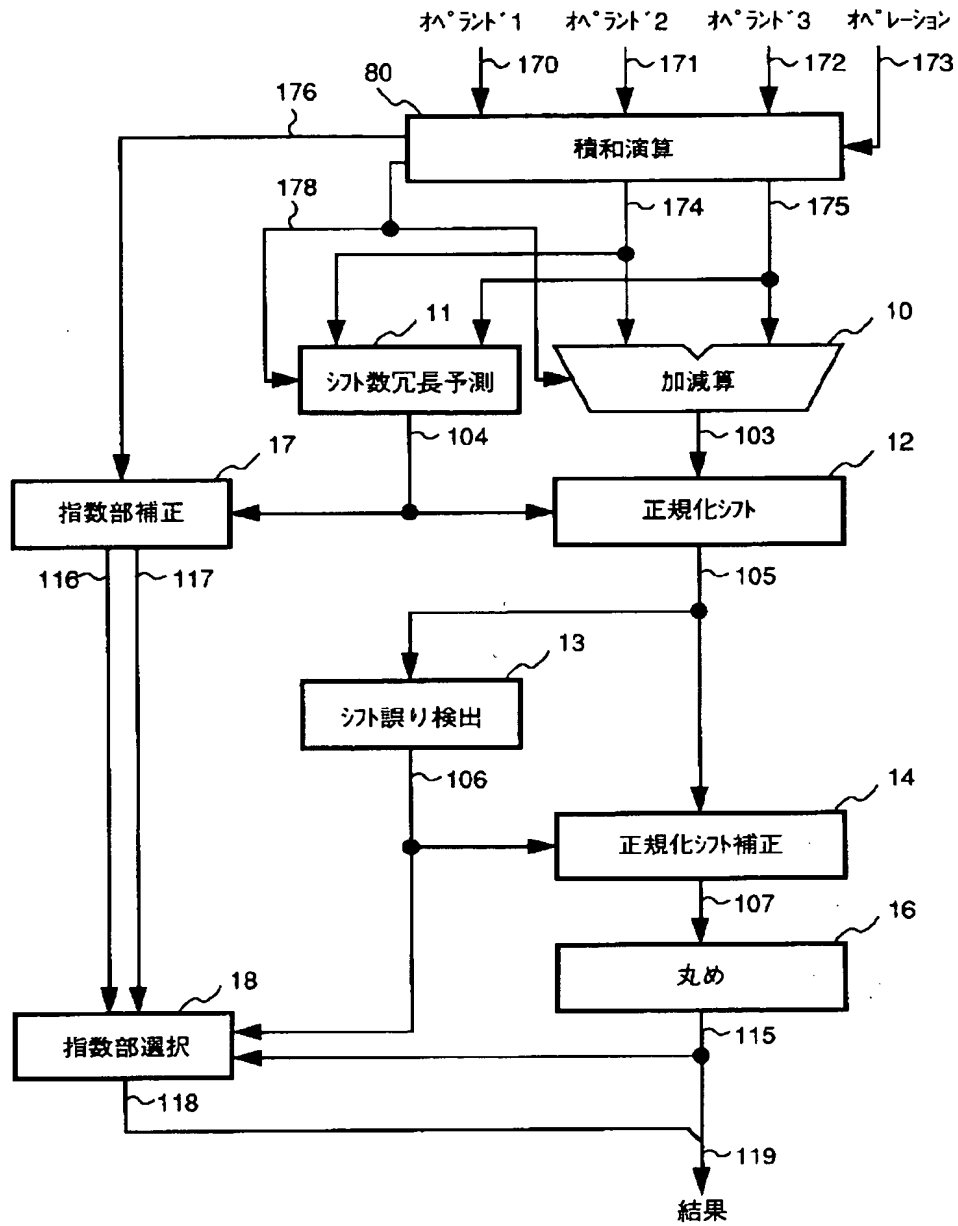
(b) 本発明の演算装置の処理時間





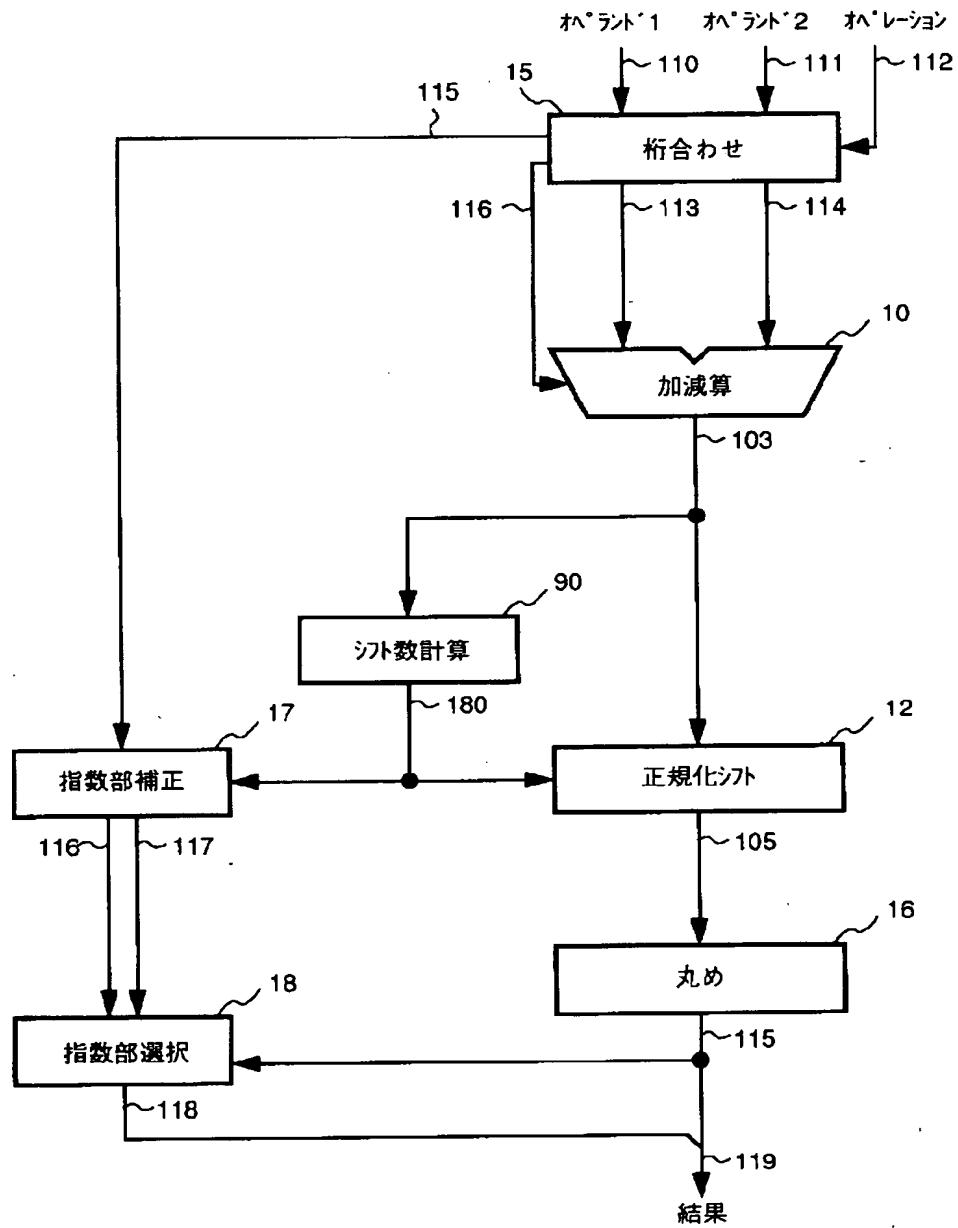
【図 15】

図 15



【図 16】

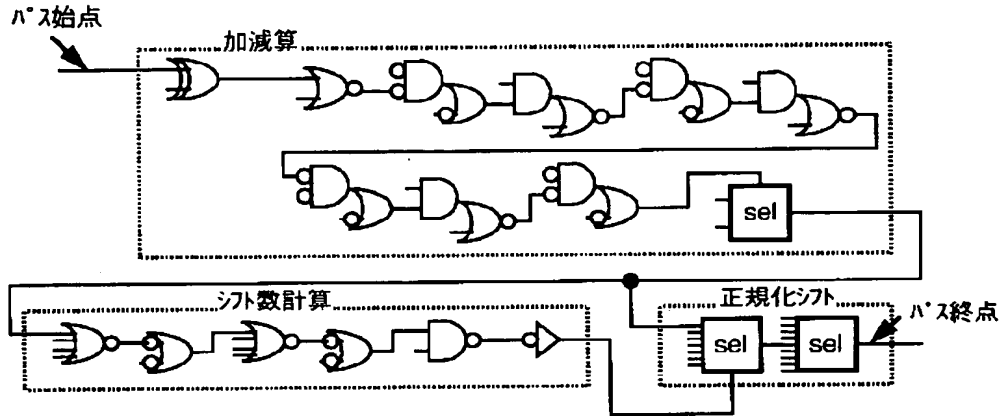
図 16



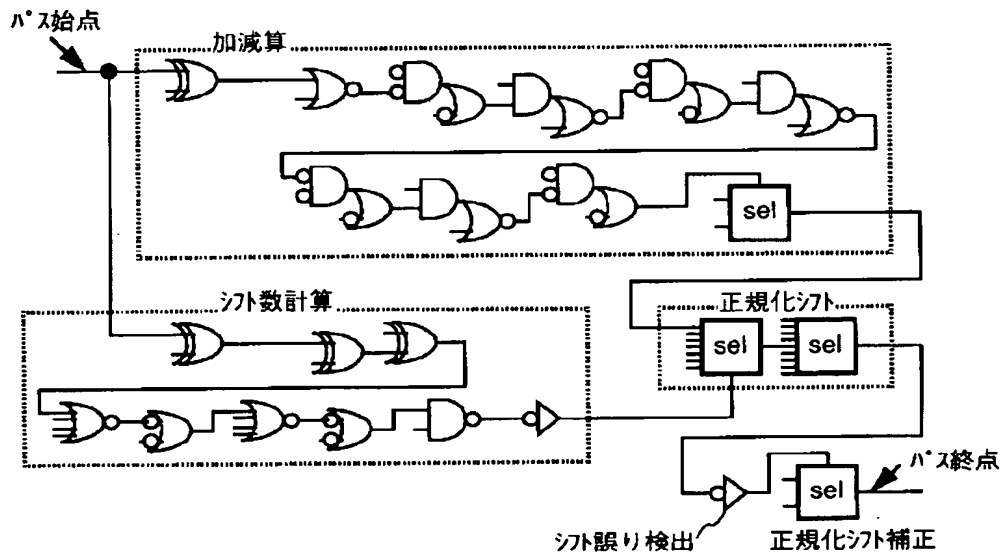
【図 17】

図 17

(a) 従来の演算装置のクリティカルパス(加減算と正規化シフト部分)



(b) 本発明の演算装置のクリティカルパス(加減算と正規化シフト部分)



フロントページの続き

(51) Int. Cl. 6

識別記号

庁内整理番号

F I

技術表示箇所

G 0 6 F 17/10

(72) 発明者 堀田 多加志  
 茨城県日立市大みか町七丁目 1 番 1 号 株  
 式会社日立製作所日立研究所内

(72) 発明者 澤本 英雄  
 神奈川県秦野市堀山下 1 番地 株式会社日  
 立製作所汎用コンピュータ事業部内

(72) 発明者 西山 隆裕  
神奈川県秦野市堀山下 1 番地 株式会社日  
立製作所汎用コンピュータ事業部内